FTML Project

nelson.vicel-farah, antoine.zellmeyer

June 2022

1 Bayes estimator and Bayes risk

Question 1

- Input space $\mathcal{X} = [0; 1]$
- Output space $\mathcal{Y} = \mathbb{R}^+$
- X uniform continuous distribution on $\mathcal{X}$
- $l(x, y) =$ squared loss
- Y $\sim$ Exp(1 + X)

The Bayes estimator in respect to the squared loss is $f^*(x) = E[Y|X = x]$, so

$$f^*(x) = E[Y \sim Exp(1 + x)]$$

We know that $E[X \sim Exp(\lambda)] = \frac{1}{\lambda}$ so

$$\mathbf{f^*(x) = \frac{1}{1 + x}}$$

And the Bayes Risk is

$$R^* = E[l(Y, f^*(X))]$$
$$= E_X[E_Y[(Y - f^*(X))^2|X]]$$
$$= E_X[Var(Y|X)]$$

We know that $Var[X \sim Exp(\lambda)] = \frac{1}{\lambda^2}$

$$= E_X[\frac{1}{(1 + X)^2}]$$

$$= \int_0^1 \frac{1}{(1 + x)^2} dx = \left[-\frac{1}{1 + x}\right]_0^1$$

$$= \mathbf{\frac{1}{2}}$$

Question 2

Let's define $\tilde{f}$ as the OLS estimator, which means

$$\tilde{f} = \begin{cases} \mathcal{X} \to \mathcal{Y} \\ x \to \hat{\theta}_1 x + \hat{\theta}_2 \end{cases}$$

With $\hat{\theta}_1$ and $\hat{\theta}_2$ the scalar parameters that minimize the squared loss. We deduce that

$$\begin{aligned}
\hat{\theta}_1 &= \frac{Cov(X,Y)}{Var(X)} \\
&= 12 * Cov(X,Y) \\
&= 12\,(E[XY] - E[X]E[Y]) \\
&= 12\,(E[E[XY|X]] - E[X]E[E[Y|X]]) \\
&= 12\,(E[\mathbf{XE[Y|X]}] - E[X]E[E[Y|X]]) \\
&= 12 \left(E[\frac{X}{1+X}] - E[X]E[\frac{1}{1+X}] \right) \\
&= 12 \left(1 - log(2) - \frac{1}{2} log(2) \right) \\
&= \mathbf{12 - 13log(2)}
\end{aligned}$$

And

$$\begin{aligned}
\hat{\theta}_2 &= E[Y] - \hat{\theta}_1 E[X] \\
&= E[E[Y|X]] - \hat{\theta}_1 \frac{1}{2} \\
&= log(2) - \frac{1}{2}(12 - 13log(2)) \\
&= \mathbf{\frac{15}{2} log(2) - 6}
\end{aligned}$$

We conclude that

$$\tilde{f}(x) = (12 - 13log(2))x + \frac{15}{2} log(2) - 6$$

The simulation in `part1_simulation.py` with the above settings and 10 000 samples of (X,Y) suggests that the Bayes estimator is better at estimating the setting than the OLS estimator. Probably because our model is not linear. Furthermore, the computed generalization error of the Bayes estimator corresponds to the value of the Bayes Risk we found previously ($\sim 1/2$) which lets us think that the simulation is a good approximation of the theoretical setup.

2 Bayes risk with absolute loss

Question 1

P(Y|X=x) where P corresponds to an Exp(λ) continuous distribution.
The Bayes estimator for l_2 squared loss

$$f_2^*(x) = E[Y|X = x] = E[\lambda e^{-\lambda}] = \frac{1}{\lambda}$$

The Bayes estimator for l_1 absolute loss is the median of Y|X=x as seen in Question 2

$$f_1^*(x) = \frac{ln(2)}{\lambda}$$

These Bayes estimators are not equal.

Question 2

We note $p = p_{Y|X=x}$

$$g(z) = \int_{\mathbb{R}} |y - z| p(y) dy$$

$$= \int_z^{+\infty} (y - z) p(y) dy + \int_{-\infty}^{z} (z - y) p(y) dy$$

$$= \int_z^{+\infty} y p(y) dy - z \int_z^{+\infty} p(y) dy + z \int_{-\infty}^{z} p(y) dy - \int_{-\infty}^{z} y p(y) dy$$

$$\frac{d}{dz} g(z) = -zp(z) - \left(\int_z^{+\infty} p(y) dy - zp(z) \right) + \left(\int_{-\infty}^{z} p(y) dy + zp(z) \right) - zp(z)$$

$$= \int_{-\infty}^{z} p(y) dy - \int_z^{+\infty} p(y) dy$$

Thus, $\frac{d}{dz} g(z) = 0$ if :

$$\int_{-\infty}^{z} p(y) dy = \int_z^{+\infty} p(y) dy$$

This occurs when both sides are equal to $\frac{1}{2}$, which means that the Bayes estimator $f^*(x)$ is the median. But to confirm this minimizes g(z), let's check the second derivative:

$$\frac{d^2}{dz^2} g(z) = 2p(z) > 0$$

The second derivative is always positive, so our solution is a minimum.

3 Expected value of empirical risk

Exercise

Step 1 :

$$E\left[R_n(\hat{\theta})\right] = E\left[\frac{1}{n}||y - X\hat{\theta}||_2^2\right]$$

$$= E\left[\frac{1}{n}||y - X((X^TX)^{-1}X^Ty)||_2^2\right]$$

$$= E\left[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)y||_2^2\right]$$

$$= E\left[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)(X\theta^* + \epsilon)||_2^2\right]$$

$$= E\left[\frac{1}{n}||X\theta^* + \epsilon - X\underbrace{(\mathbf{X^TX})^{-1}\mathbf{X^TX}}_{=\mathbf{I_d}}\theta^* - (X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

$$= E\left[\frac{1}{n}||X\theta^* + \epsilon - X\theta^* - (X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

$$= E\left[\frac{1}{n}||\epsilon - (X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

$$= E_\epsilon\left[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

Step 2 :

$$A \in \mathbb{R}^{n,m} B \in \mathbb{R}^{n,m}$$

$$tr(A^TB) = \sum_{i,j \in [1,n]\times[1,m]} a_{ij}b_{ij}$$

Thus :

$$A \in \mathbb{R}^{n,n}$$
$$tr(A^TA) = \sum_{i,j} a_{ij}a_{ij}$$

Step 3 :

$$E_\epsilon\left[\frac{1}{n}||A\epsilon||^2\right] = E_\epsilon\left[\frac{1}{n}\sum_{i=1}^{n}(A\epsilon)_i^2\right] = E_\epsilon\left[\frac{1}{n}\epsilon^2\sum_{i,j\in[1,n]^2}A_{ij}^2\right]$$

$$= E_\epsilon\left[\frac{1}{n}\epsilon^2 tr\left(A^TA\right)\right]$$

$$= \frac{1}{n}tr(A^TA)E_\epsilon[\epsilon^2]$$

$$= \frac{1}{n}tr(A^TA)(E_\epsilon[\epsilon^2] - \underbrace{\mathbf{E}_\epsilon[\epsilon]^\mathbf{2}}_{=\mathbf{0}})$$

$$= \frac{1}{n}tr(A^TA)\sigma^2$$

Step 4 :

$$A = I_n - X(X^TX)^{-1}X^T$$

$$A^TA = (I_n - X(X^TX)^{-1}X^T)^T(I_n - X(X^TX)^{-1}X^T)$$

$$= (\mathbf{I_n^T} - (\mathbf{X}(\mathbf{X^TX})^{-\mathbf{1}}\mathbf{X^T}))^\mathbf{T})(I_n - X(X^TX)^{-1}X^T)$$

$$= (\mathbf{I_n} - (\mathbf{X^T})^\mathbf{T}((\mathbf{X^TX})^{-\mathbf{1}})^\mathbf{T}\mathbf{X^T})(I_n - X(X^TX)^{-1}X^T)$$

$$= (\mathbf{I_n} - \mathbf{X}(\mathbf{X^TX})^{-\mathbf{1}}\mathbf{X^T})(I_n - X(X^TX)^{-1}X^T)$$

$$= (I_n - X(X^TX)^{-1}X^T)^2$$

$$= I_n - 2X(X^TX)^{-1}X^T + (X(X^TX)^{-1}X^T)^2$$

$$= I_n - 2X(X^TX)^{-1}X^T + X(X^TX)^{-1}\underbrace{\mathbf{X^TX}(\mathbf{X^TX})^{-\mathbf{1}}}_{=\mathbf{I_d}}X^T$$

$$= I_n - 2X(X^TX)^{-1}X^T + X(X^TX)^{-1}X^T$$

$$= I_n - X(X^TX)^{-1}X^T = \mathbf{A}$$

Thus : $\mathbf{A^TA = A}$

Step 5 : Conclude

$$
\begin{aligned}
E\left[R_n(\hat{\theta})\right] &= E_\epsilon\left[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)\epsilon||_2^2\right] \\
&= \frac{\sigma^2}{n} tr(I_n - X(X^TX)^{-1}X^T) \\
&= \frac{\sigma^2}{n}\left(n - tr(X(X^TX)^{-1}X^T)\right) \\
&= \frac{\sigma^2}{n}\left(n - tr(X^TX(X^TX)^{-1})\right) \\
&= \frac{\sigma^2}{n}\left(n - tr(I_d)\right) \\
&= \frac{\sigma^2}{n}(n-d)
\end{aligned}
$$

Thus :

$$E[R_X(\hat{\theta})] = E[E[R_n(\hat{\theta})]] = E[\frac{n-d}{n}\sigma^2] = \frac{n-d}{n}\sigma^2$$

Simulation

Step 6

$$E\left[\frac{||y - X\hat{\theta}||_2^2}{n-d}\right]$$

We recognize the fixed design risk with a subtle difference : $\frac{1}{n-d}$ replaced $\frac{1}{n}$, so we can use the formula found in Step 5 and apply this change.

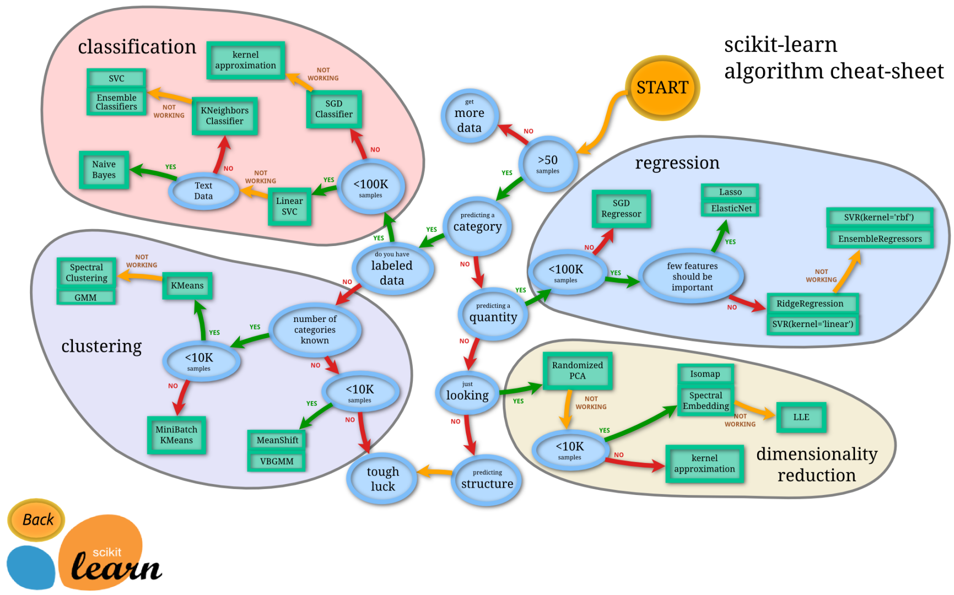
$$= \frac{n-d}{n-d}\sigma^2 = \sigma^2$$

Step 7

The simulation is in `part3_step7.py`. We suppose : n = 10000, d = 20, $y = X\theta^*$ with θ^* a random vector. We thus estimate σ^2 with the result of Step 6. With this setting, the simulation estimates a σ^2 of 0, which is an expected result knowing that y is a simple linear function of X without any noise added to it (so the variance of ϵ is 0).
In a second setting, we define y as $y = X\theta^* + \epsilon$ with ϵ being a gaussian vector with a standard deviation of 2 and a mean of 0, for which the simulation estimates σ^2 as 4, thus being still consistent with the theoretical values.

4 Regression

For this section and the next one, we chose `scikit-learn` as our machine learning library because it contains many regression and classification models and allows us to follow this decision tree :



We should then retrieve the important properties of the dataset :
Number of entries : 1000
Number of features : 20
This needs a regression model that can handle a small set of samples and 20 features

Model	R2 score
SVR Linear	0.88757
SVR rbf	0.66461
Ridge alpha=12.0	**0.89021**
Ridge alpha=15.0	**0.89027**
Lasso alpha=12.0	0.83622
RandomForestRegressor	0.82288
AdaBoost Regressor	0.82405
Gradient Boosting Regressor	0.86165

Although we should note that the R2 score is highly influenced by the random initialization of the models parameters, the Ridge regression seems to be globally the best regression model in this case.

5 Classification

As we did in the previous section, we retrieve useful information about the dataset
Number of entries : 1000
Number of features : 20
This needs a classification model that can handle a small set of samples and 20 features. Let's try some of them:

Model	Accuracy
LinearSVC	**0.916**
KNeighbors	0.864
SVC	0.892
RandomForestClassifier	0.876
AdaBoostClassifier	0.88
GradientBoostingClassifier	0.868

Although we should note that the accuracy is highly influenced by the random initialization of models parameters, the linear SVC seems to be globally the best classification model in this case.